



Reachability Questions? Developers Love This One Weird Tool

James Yoo **MSc** Thesis Presentation October 17, 2022

THE UNIVERSITY OF BRITISH COLUMBIA SOFTWARE PRACTICES LABORATORY





	? How was model created?
File	e Directory Project
✓ 150	VirtualFile directory = FindInProjectUtil.getDirectory(model); in FindPopupDirectoryChooser.validate(FindMode
\sim	149 public ValidationInfo validate(@NotNull FindModel model) { in FindPopupDirectoryChooser.validate(FindModel model) {
	✓ 168 return myDirectoryChooser.validate(model); in FindPopupScopeUIImpl.validate(FindModel, ScopeType) (
	166 public ValidationInfo validate(@NotNull FindModel model, FindPopupScopeUI.ScopeType selectedSco
	1479 ValidationInfo scopeValidationInfo = myScopeUI.validate(findModel, mySelectedScope); in FindPo
	1477 FindModel findModel = new FindModel(); in FindPopupPanel.getValidationInfo()
1475	<pre>@Nullable("null means OK")</pre>
1476	<pre>private ValidationInfo getValidationInfo() {</pre>
1477	FindModel findModel = <mark>new FindModel()</mark> ;
1478	<pre>this.directory = findModel.getDirectoryName();</pre>
1479	ValidationInfo scopeValidationInfo = myScopeUI.validate(findModel, mySelectedScop
1480	<pre>if (scopeValidationInfo != null) {</pre>



Developers Developers Developers What do they do, *really*?





2





Amy J. Ko, Htet Aung, and Brad A. Myers. 2005. Eliciting design requirements for maintenance-oriented IDEs: a detailed study of corrective and perfective maintenance tasks. In Proceedings of the 27th international conference on Software engineering (ICSE '05)

20% Editing code

Amy J. Ko, Htet Aung, and Brad A. Myers. 2005. Eliciting design requirements for maintenance-oriented IDEs: a detailed study of corrective and perfective maintenance tasks. In Proceedings of the 27th international conference on Software engineering (ICSE '05)







Amy J. Ko, Htet Aung, and Brad A. Myers. 2005. Eliciting design requirements for maintenance-oriented IDEs: a detailed study of corrective and perfective maintenance tasks. In Proceedings of the 27th international conference on Software engineering (ICSE '05)





Amy J. Ko, Htet Aung, and Brad A. Myers. 2005. Eliciting design requirements for maintenance-oriented IDEs: a detailed study of corrective and perfective maintenance tasks. In Proceedings of the 27th international conference on Software engineering (ICSE '05)

52%

Program comprehension activities



7

What Questions Do Developers Ask?

- Survey of 460 software engineers at Microsoft
- Developers ask *reachability questions*
 - Asked them more than 9 times a day
 - 82% of them reported them as being somewhat hard to answer

Field study of 17 developers

Reachability questions accounted for 9 of the 10 longest activities

Thomas D. LaToza and Brad A. Myers. 2010. Developers ask reachability questions. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE '10).



Developers Ask Reachability Questions

LaToza and Myers' Framework

find SC in TR

- Looking through traces to find points of interest
- "Where is this method generating an error?"
- "How is data being mutated between these two method calls?"



LaToza and Myers' Framework

find SC in TR

- Looking through traces to find points of interest
- "Where is this method generating an error?"
- "How is data being mutated between these two method calls?"

 $compare(TR_a, TR_b) : TR_{common}, TR_1, TR_2$

- Given two traces, find what is common, and what is unique to each trace
- "What's the difference between the test and production version of this app?



REACHER [1]

- Focuses on control-flow graph exploration
- Invoked via menu

Get Me Here [2]

- Reframes reachability questions to SMT problems
- Invoked via menu
- No user study on usability

[1] Thomas D. LaToza and Brad A. Myers. 2010. Visualizing call graphs. 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) [2] Mike Barnett et al. Get Me here: using verification tools to answer developer questions. Microsoft Research Technical Report 2014-10



Reachability Questions in Practice

RQ1 What are the reachability questions frequently encountered by developers in practice?

Reachability Questions in Practice

RQ1 What are the reachability questions frequently encountered by developers in practice?

Survey of 72 practicing software developers

- 9 reachability questions from the literature, past experiences
- Distributed via Twitter

This figure represents a scenario where some data is created (the cart object), and and passed along as an argument to some methods. It may not be immediately clear how this data is being **modified** (e.g., which fields of the **cart** object are being updated or changed).



How often do you ask yourself this question: Given some data (in this case, cart), which parts of it are modified downstream?

> I never ask myself this question 2

		/ '	
lon	hui	1 d(۱.
ier.	DOT	LUC	,,

Are you interested in how cart could be modified in these method calls?

throw new CartValidationException("Provisioning a new cart failed");

I often ask myself this question

3 5

Reachability Questions in Practice

RQ1 What are the reachability questions frequently encountered by developers in practice?

Survey of 72 practicing software developers

- 9 reachability questions from the literature, past experiences
- Distributed via Twitter

Reachability Questions in Practice

RQ1 What are the reachability questions frequently encountered by developers in practice?

Survey of 72 practicing software developers

- 9 reachability questions from the literature, past experiences
- Distributed via Twitter

Developers frequently encounter reachability questions that are related to the data-flow of a program during their work

ReachHover



			▶3	
		? How	v was model created?	
		File Directory Project		
Fo validate(GNotNull FindModel model) {	150 VirtualFile directory = FindInProjectUtil.getDire	ectory(model); in FindPopupDirectoryChooser.validate(FindModel)	
<pre>tory = FindInProjectUtil.getDirectory(model):</pre>		 149 public validationinio validate(@Notivul FindModel model) { in FindPopupDirectoryChooser.validate(FindModel) 168 return mvDirectoryChooser.validate(model): in FindPopupScopeUIImpl.validate(FindModel, ScopeType) (filter: 		
How was model created?		 166 public ValidationInfo validate(@NotNut	Ill FindModel model , FindPopupScopeUI.ScopeType selectedScope) { = myScopeUI.validate(findModel , mySelectedScope); in FindPopupF	
	Show types and documentation?	1477 FindModel findModel = new Fi	indModel(); in FindPopupPanel.getValidationInfo()	
		1475 <a>(ONullable("null means OK") 1476 <a>private ValidationInfo getValidati	onInfo() {	
		= 1477 FindModel findModel = new FindModel	del();	
		1478 this.directory = findModel.getDi	rectoryName();	
		1479 ValidationInfo scopeValidationIn	<pre>fo = myScopeUI.validate(findModel, mySelectedScope);</pre>	
		1480if (scopeValidationInfo != null)1481return scopeValidationInfo;	-{	
1476	<pre>private ValidationInfo getVa</pre>	alidationInfo() {		
1477	FindModel findModel = <mark>new</mark>	<pre>FindModel();</pre>	platform/lang-impl/src/com/intellij/find/impl	
1478	<pre>this.directory = findModel</pre>	L.getDirectoryName();		

Daniil Ovchinnikov

```
@Nullable
```

```
public ValidationInfo validate(@NotNull FindModel model) {
 VirtualFile directory = FindInProjectUtil.getDirectory(model);
 if (directory = null) {
  }
 return null;
}
```

return new ValidationInfo(FindBundle.message(key: "find.directory.not.found.error"), myDirectoryComboBox);

💄 Daniil Ovchinnikov

@Nullable

```
public ValidationInfo validate(@NotNull FindModel mo
VirtualFile directory = FindInProjectUtil.getDirect
if (directory = null) {
    return new ValidationInfo(FindBundle.message( kee
    }
    return null;
}
```

odel) {		
<i>ctory</i> (mode	el);	
	How was model created?	
ey: "find.d	Show types and documentation?	<pre>myDirectoryComboBox);</pre>

	F	ile Directo
	✓ 15✓	0 VirtualFile d 149 public Va 168 return 166 pul 1479
🚨 Daniil Ovchinnikov		14
0Nullable	1465	@Override
nublic VolidationInfo volidato(@NotNull Fi	1466	@Nullable
public validationinto validate(@Nothult Fi	1467	public Str
VirtualFile directory = FindInProjectUt	1468	String <u>n</u>
<pre>if (directory = null) {</pre>	1469	if (head
noturn new VolidationInfo(EindBundlo	1470	<u>mask</u> =
recorn new vacidationinfo(FindBondle.n	1471.	}
}	1472	return <u>n</u>
return null;	1473	}
1	1474	QNullahla
l	1475	private Va
	1477	
	1478	this.dir
	1479	Validati
	1480	if (scop
	1481.	returr
	1482	}
	1483	
	1484	if (!myŀ
	1485	returr
	1486	}
	🖵 Fir	ndPopupPanel

How was model created?

ectory Project

le directory = FindInProjectUtil.getDirectory(**model**); in FindPopupDirectoryChooser.validate(FindModel) c ValidationInfo validate(@NotNull FindModel **model**) { in FindPopupDirectoryChooser.validate(FindModel) curn myDirectoryChooser.validate(**model**); in FindPopupScopeUIImpl.validate(FindModel, ScopeType) (filter: not-r public ValidationInfo validate(@NotNull FindModel **model**, FindPopupScopeUI.ScopeType selectedScope) { in Fin 479 ValidationInfo scopeValidationInfo = myScopeUI.validate(**findModel**, mySelectedScope); in FindPopupPanel. 1477 FindModel findModel = **new FindModel()**; in FindPopupPanel.getValidationInfo()

```
ide
ble
String getFileTypeMask() {
ng mask = null;
header.cbFileFilter ≠ null && header.cbFileFilter.isSelected()) {
sk = (String)header.fileMaskField.getSelectedItem();
rn mask;
ble("null means OK")
e ValidationInfo getValidationInfo() {
Model findModel = new FindModel();
.directory = findModel.getDirectoryName();
dationInfo scopeValidationInfo = myScopeUI.validate(findModel, mySelectedScope);
scopeValidationInfo;
!myHelper.canSearchThisString()) {
```

urn new ValidationInfo(FindBundle.message("find.empty.search.text.error"), mySearchComp

platform/lang-impl/src/com/intellij/find/impl

The End







Developer-centric needs What's missing?







Developer-centric needs What's missing?







Developer-centric needs What's missing?

Evaluation Did we build the right thing?



- **RQ2** Are developers able to answer data-flow reachability questions more correctly with ReachHover?
- **RQ3** Does ReachHover make it easier to answer data-flow reachability questions?

- **RQ2** Are developers able to answer data-flow reachability questions more correctly with ReachHover?
- **RQ3** Does ReachHover make it easier to answer data-flow reachability questions?

Controlled experiment with 20 software developers

- Two tasks posing data-flow reachability questions
- Remote study with on-device logging

What are we measuring? Correctness

Scoring answers to reachability questions

Visual Momentum

Editor cursor clicks and jumps, proxy for context switching

Developer Experience

Open coding on answers to two experiential questions

 $S = ANS_{correct} - ANS_{incorrect}$

Task Example How is a value created?

1.	In how many locations	
	does a call to the	341
	BookmarkInfo constructor	342
	exist?	343

344 @

350

351

352

}

- 345 2. Provide the names of the 346 methods where the value 347 of **bookmark** is used 348 349
- 3. Provide the names of the methods where **bookmark** is assigned a value

```
/**
 * Add a new bookmark
 */
2 usages 💄 James Yoo
public void addBookmark(FileBookmarks fileBookmarks, BookmarkInfo bookmark) {
    checkModDuringFire();
    fileBookmarks.add(bookmark);
    BookmarkChange change = getBookmarkChange(bookmark);
    change.markAdded();
    bookmark.getFileBookmarks().getProjectBookmarks().setModified(true);
    structureChange = true;
```



Average Correctness Score Backward, inter-file reachability question

- ReachHover: 4.1
- Built-in Data-flow: **0.8**



Average Correctness Score Forward, intra-file reachability question

- ReachHover: 4.6
- Built-in Data-flow: 4.1



Visual Momentum

- Fewer cursor jump actions in the main editor on average for each task
- Statistically significant difference for the forward intra-file task

(Mann-Whitney U = 6, n = 10, p = 0.001, a = 0.05)



Developer Experience

EQ1 Did you find any difference usi this study?

Did you find any difference using ReachHover or IntelliJ for the tasks in

Developer Experience

- EQ1 this study?
- **19 total responses**
 - **13** (65%) reported a positive experience with ReachHover
 - Easier and simpler to use (5)
 - Helps maintain context (5)
 - More usable (3)



Did you find any difference using ReachHover or IntelliJ for the tasks in

• Some reported no difference (4), while others (2) preferred the built-in IDE features

"ReachHover felt less cluttered and made the task easier" – P18

"ReachHover was easier to call as it didn't require navigating menus" – P9

"I find having ReachHover's separate floating windows with a view into where the related line of code is, is more usable than IntelliJ's data-flow analysis feature" – P10

"ReachHover felt less cluttered and made the task easier" – P18

"ReachHover was easier to call as it didn't require navigating menus" – P9

"I find having ReachHover's separate floating windows with a view into where the related line of code is, is more usable than IntelliJ's data-flow analysis feature" – P10

"I preferred that I could look at the matches in the same modal window without affecting the open file" – P19

"I felt like I did not lose the context using ReachHover compared to IntelliJ. I felt they showed me the same content, but ReachHover was easier to use (maybe beause [sic] I didnt [sic] lose the context)" – P15

"I preferred that I could look at the matches in the same modal window without affecting the open file" – P19

"I felt like I did not lose the context using ReachHover compared to IntelliJ. I felt they showed me the same content, but ReachHover was easier to use (maybe beause [sic] I didnt [sic] lose the context)" – P15

RQ2 Are developers able to answer data-flow reachability questions more correctly with ReachHover?

RQ3 Does ReachHover make it easier to answer data-flow reachability questions?

ReachHover enabled participants to answer an interfile data-related reachability question more accurately and an intra-file question as accurately, as built-in data-analysis support in an IDE

RQ3 Does ReachHover make it easier to answer data-flow reachability questions?

ReachHover enabled participants to answer an interfile data-related reachability question more accurately and an intra-file question as accurately, as built-in data-analysis support in an IDE

We observed a lower average of context-shifting main editor interactions when ReachHover was used compared to built-in IDE support. The majority of our participants found ReachHover to be easier or simpler to use, more usable, and helpful in preserving their context during tasks.

Discussion and Future Work

Discussion and Future Work

Asking and Answering Additional Reachability Questions

- Investigate compare-type questions
- Run-time information for dynamic reachability questions
- Use test execution data, production logs, dynamic slicing

